University of Southern California

2

DRAMA:
An Application of a Logistic Shell

Brian Harp
Peter Aberg
David Benjamin
Robert Neches
Pedro Szekely

March 1991
ISI/RR-91-284

INFORMATION
SCIENCES
INSTITUTE
ISI
213/822-1511
4676 Admiralty Way/Marina del Rey/California 90292-6695

DRAMA:
An Application of a Logistic Shell

Brian Harp
Peter Aberg
David Benjamin
Robert Neches
Pedro Szekely

March 1991
ISI/RR-91-284

REPORT DOCUMENTATION PAGE

1a. REPORT SERCURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILTY OF REPORT

4. PERFORMING ORGANIZATION REPORT NUMBER(s)
ISI/RR-90-284

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
USC
INFORMATION SCIENCES INSTITUTE

6b. OFFICE SYMBOL

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)

4676 Admiralty Way
Marina del Rey, California 90292

7b. ADDRESS (City, State, and Zip Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
DARPA and DLA-OSS

8b. OFFICE SYMBOL

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

3701 Fairfax Drive
Arlington, Virginia 22203

and

Cameron Station
Alexandria, VA 22304

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (INCLUDE SECURITY CLASSIFICATION)

DRAMA: AN APPLICATION OF LOGISTICS SHELL

12. PERSONAL AUTHOR(S)

Brian Harp, Peter Aberg, David Benjamin, Robert Neches, Pedro Szekely

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF EPORT (Year, Month,Day)	15. PAGE COUNT
Research Report	FROM ________ TO ________	1991, March	six

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

Logistics Support, Knowledge Based System

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Managing the logistics of system repair and maintenance is a demanding process in bot the government and commercial sector. It requires interpretation and management of large amounts of highly specialized information. Current software technology and computer interfaces are not sufficient for the growing demands of logistcs support and so new approaches are being pursued. Information Sciences Institute (ISI) is actively developing new technology to face future demands and has developed a framework for building logistical support applications.

DRAMA (Data Review, Analysis and Monitoring Aid), an application being developed using this framework addresses item entry and control of new weapon systems being introduced into the U.S. military. DRAMA is a part of the solution to the growing demands on the logistics support in the government. It monitors and analyzes support information available for weapon systems and assists personnel (e.g., catalogers, Item Managers) in making weapon systems support more effective and efficient. This paper describes a framework called "logistic shell" and DRAMA'S functionality based on this framework.

20. DISTRIBUTION/AVAILBILIT OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b TELEPHONE (Include Area Code)
(310) 822-1511

22c. OFFICE SYMBOL

DRAMA: An Application of a Logistics Shell

Brian Harp
Peter Aberg, David Benjamin, Robert Neches, Pedro Szekely

USC/Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
213.822.1511

1.0 Overview

Managing the logistics of system repair and maintenance is a demanding process in both the government and commercial sector. It requires interpretation and management of large amounts of highly specialized information. Current software technology and computer interfaces are not sufficient for the growing demands of logistics support and so new approaches are being pursued. Information Sciences Institute (ISI) is actively developing new technology to face future demands and has developed a framework for building logistical support applications.

DRAMA[1] (Data Review, Analysis and Monitoring Aid), an application being developed using this framework, addresses item entry and control of new weapon systems being introduced into the U.S. military. DRAMA is a part of the solution to the growing demands on the logistics support in the government. It monitors and analyzes support information available for weapon systems and assists personnel (e.g., catalogers, Item Managers) in making weapon systems support more effective and efficient. This paper describes a framework called a "logistics shell" and DRAMA's functionality based on this framework.

2.0 The Purpose of DRAMA

DRAMA is an application based on the IN-USE framework [3], a generic framework for constructing sophisticated data management systems. This framework provides software modules which assist in inspecting data, in monitoring data for changes, in recording user- or system-generated notes concerning that data, and in planning and executing user- or system-initiated actions. This framework has been found to be useful in building logistics applications such as DRAMA.

[1] This research is being supported by DARPA in cooperation with DLA under contract #53-4540-9402. The contents of this paper represent the views and conclusions of the authors, and do not necessarily reflect the policies or beliefs of the sponsoring agencies.

DRAMA has been constructed by extending the modules provided in this framework, and impacts weapon system support in two areas; (1) it starts managing information about a weapon system earlier in the life cycle than is currently done and (2) it assists Item Managers in tracking changes to weapon systems which affect supportability of those weapon systems. These two improvements are achieved by accessing databases that have not been utilized before, by using tools and inference techniques which are knowledge intensive, and by providing an interface that provides access to this system knowledge in natural ways.

3.0 The Logistics Shell

The logistics shell consists of a number of loosely-coupled modules and databases that can be combined in various ways to build new systems. Each module roughly consists of a knowledge base (KB) model and a set of inference techniques which operate using that model. For example, one sub-module,The Intelligent Note Taker (TINT), provides capabilities to attach free text to objects in the knowledge bases. In the logistics shell, TINT consists of a model containing generic note types (e.g., User-created-notes, Activity-notes) and functions for note attachment, retrieval and inference.

When a developer uses the logistics shell to implement a new application like DRAMA, they extend the modules provided, link them together through shared knowledge bases and create an interface which is suitable to the end users. Extending a module requires extending the knowledge bases to contain domain-specific information and relating concepts in the knowledge bases to data in the domain databases. It also requires the developer implement the application-specific functionality of the module by utilizing existing, domain-independent functionality and developing new functionality if necessary.

There are three primary modules provided in the logistics shell. The *Incoming Data Assimilation* module provides the interface between the local knowledge bases and any databases that are to be accessible through the application. In applications, this module is extended to retrieve data from the databases and convert it to the representation required for local reasoning in the *Data Review and Analysis* module. The Data Review and Analysis module (the second primary module) contains a knowledge base and associated functionality to identify patterns in the data that are anomalous or require some action be taken. It also contains a module that manages tasks created in response to anomalies and capabilities to interface with external systems. The third module is a *User Interaction* module that provides user interface capabilities for an application..

4.0 The DRAMA Architecture

The DRAMA architecture (shown in Figure 1) utilizes all the major modules in the logistics shell. In addition, DRAMA required a module called *LSAR Processing* which converts data from a flat file format into a database representation and asserts the data into the database. The primary modules in DRAMA have essentially the same functionality as those provided in the logistics shell, except that the knowledge bases in each module have been extended to contain additional, DRAMA-specific information. Following are descriptions of each module.

5.0 Module Description and Functionality

5.1 LSA Files and database

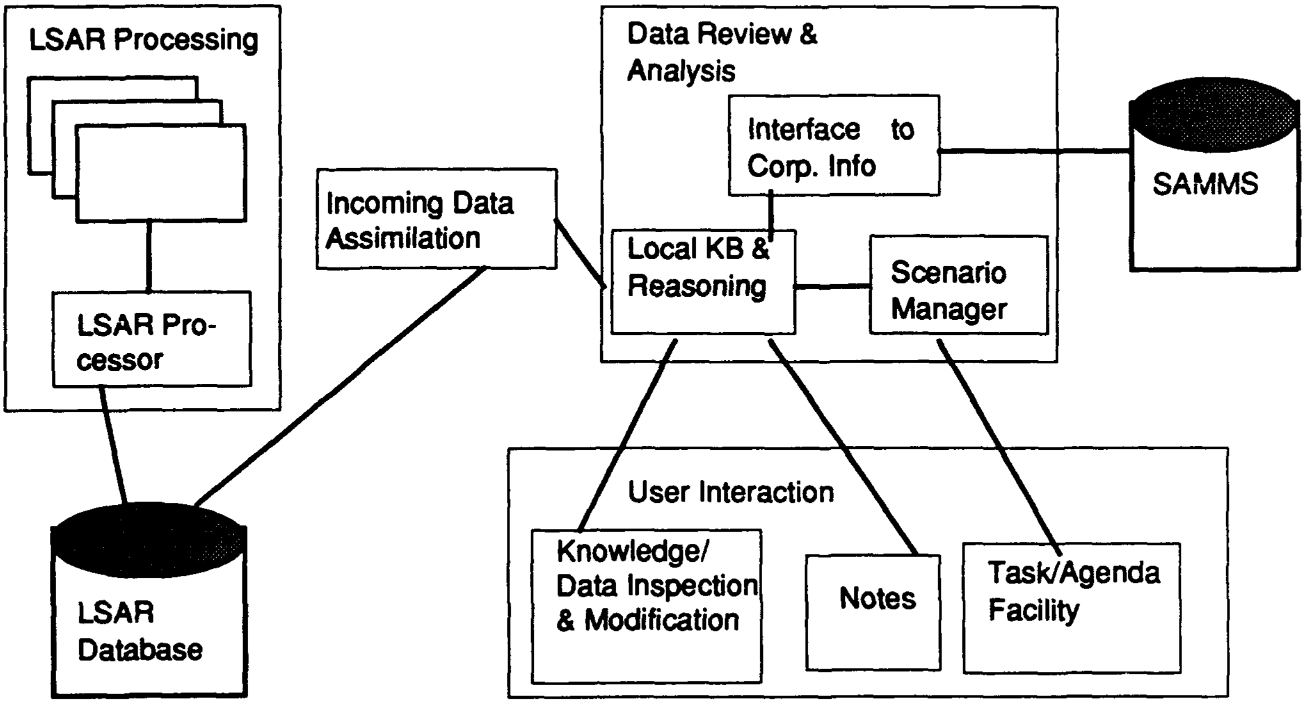

Fig. 1: The DRAMA Architecture

As was mentioned above, LSAR data is received in flat files. These files are currently formatted as reports defined by Mil-Std #1388. LSAR files contain about seven hundred attributes for each part on a weapon system describing aspects of the part such as stockage location, failure rates and material content. Since DRAMA utilizes relational database systems for storing this data, it reads the LSAR files, converts them into database tuples and stores the data in the LSAR database.

5.2 Incoming Data Assimilation

This subsystem manages interactions between the database and the local KB and maintains information on what to monitor in the database. Because there is a difference in data representation between the relational database (relational tuples) and the local KB (frames), this module converts the data to the appropriate representation during the transaction. The converted data is asserted into the local KB so analysis of the data can be performed. Data is swapped in and out of local memory as required since there is too much data to reason about all at once.

6.0 Data Review and Analysis

This subsystem does much of the core processing performed in logistics support applications. The knowledge base provided by this module contains concept hierarchies defining entities such as *anomaly patterns* (concepts that indicate something out of the ordinary), *notes* (objects relating free text to KB objects) and *scenarios* (objects representing tasks to be performed). Information represented in DRAMA's local knowledge base which extends these generic models includes a model of the LSAR data, a model of anomalies found in DRAMA, a model of the transactions used in DRAMA (e.g., Supply Support requests, Recommended buys) and a model of the note types that are used in DRAMA(e.g., LSAR notifications).

When DRAMA receives a database update, or new transactions, it asserts the data into the local knowledge base and classifies it to detect anomalies based on the defined anomaly patterns. For example, one anomaly pattern represents the concept of "an LSAR item with a missing UPA (Units Per Assembly) attribute". Whenever data for an LSAR item is missing a value for the UPA field, that data will classify under that concept. After classification has taken place, the data will be linked to any applicable anomaly patterns. The Data Review and Analysis module collects the associated patterns and creates an anomaly object representing each new anomaly found. These objects are then asserted it into the local knowledge base.

The scenario manager provides a mechanism for describing and executing activities that require processing by users and/or the application software. Scenarios are program–like descriptions of the sub–tasks that compose extended tasks. Scenarios are like high–level procedures, in that they describe a sequence of steps to be performed. However, unlike a procedure, a scenario attempts to capture the processes or sequences of activities that users typically perform. Further, a scenario imposes only orderings between steps that are necessitated by dependencies between them. Finally, scenarios are constructed with pointers to data that they depend on. If this data changes, the scenario has the capability to adapt to those changes.

Utilizing scenarios in DRAMA requires that the developer define specific DRAMA scenarios. The developer must understand what anomalies activate each scenario and specify these relations when defining the scenarios. Further, the developer must be aware of how changes to data in the system affects the scenarios and what activities are required if those changes occur.

7.0 User Interaction

The User Interaction module is based on a an interface system called Humanoid [4] and provides all the user interface capabilities for the logistics shell. It supports multiple interaction paradigms and allows developers to easily define new interfaces. Although some customization of interface forms provided by the shell is always required for a new application, many of the underlying interaction paradigms are common across applications and can be reused. In DRAMA, the paradigms for browsing data were used extensively while the look and feel of DRAMA was customized to fit the user's interface requirements.

7.0.1 Knowledge / Data Inspection and Modification Facility

This module provides general capabilities for allowing users to interact with the system and control the presentation of information. Users of these large data systems typically require easy access to data through the interface and data inspection facilities related to their current focus of interest.

For any display that the user is interacting with, this module gives them three primary ways of controlling the information presentation. They can select *filters* which define the set of objects to be displayed (e.g., components with item management code Z). This module provides a set of predefined filters for each kind of object that can be presented, and provides facilities for users to create new filters. In addition to filtering, the module lets users select *data options* to determine what kind of information gets displayed about each object, and choose *orderings* that determine how objects get placed on the screen. These options also typically contain a set predefined choices and allow additional ones be added by user.

7.0.2 Notes Facility

TINT (see section 3.0 on page 2) provides an interface allowing users to enter information that may be either too detailed to represent or was unanticipated when the system was implemented. This information can be entered into the computer in the form of "semi-structured" notes [1] that are attached to objects in a knowledge base.

Even though applications like DRAMA cannot fully interpret some user-supplied notes, the computer can still operate with at least partial understanding because the notes have structure. This structure allows notes to be associated with knowledge base entries and the computer can be programmed to understand these links. When the computer encounters a semi-structured note, it can prompt the user for assistance, providing a more collaborative style of human/computer interaction than typical expert systems provide. An added benefit of allowing users to create notes is that over time, developers can analyze user notes and discover new reasoning processes that are performed routinely. These new processes can either be added to the application if they are unique, or they can be added to the general framework if they apply across logistic support applications.

7.0.3 Task / Agenda Facility

This module provides the user interface to the Scenario mechanism described above. It builds upon the capabilities provided by the core Knowledge / Data Inspection and Modification facility, while adding interactive capabilities specialized for working with displays of tasks. These include the ability to expand displays of tasks to determine their sub-tasks, the ability to ask for explanatory help and documentation about tasks, and the ability to "accelerate" execution of tasks. They also include the ability to look at collections of tasks, which are referred to as agendas, to select tasks from agendas, to get information about the status of tasks, and so on.

8.0 Conclusion

DRAMA is a powerful and flexible logistics support system providing assistance in item entry and control. It contributes in two important areas: access to more data earlier in the lifecycle of a weapon system and ongoing data monitoring. These two capabilities enable users to do a better job supporting maintenance of complex weapon systems.

DRAMA was developed by extending a logistics shell. The LSAR Processing module was integrated into the general framework with customized versions of The Data Review and Analysis module, the Incoming Data Assimilation module, and the User Interaction module. Module customization consisted of adding DRAMA-specific knowledge to the module knowledge bases and DRAMA-specific functions to each module.These four modules interact via shared information in the databases and local knowledge base.

Initially, the LSAR file processor reads data from flat files and deposits the information into the relation database. Once new data exists, the data assimilation module asserts all changes into the local knowledge base and classifies that data according to predefined patterns. If the data classifies under any patterns representing anomalies, an anomaly object is created to record it. The scenario manager is notified about each anomaly object so that any applicable activities can be executed and the anomaly can be resolved. Some tasks in these scenarios will be performed by the systems and some by the user. If there are user tasks, the corresponding scenario is linked to the agenda. Finally, if the agenda is being presented on the screen, the scenario will appear on the screen to collect user input.

The shell provides a number of useful interactive capabilities which are specialized for use in DRAMA. It provides a flexible note-taking facility that allows the user to link free text with objects in the database. The notes are semi-structured, providing an opportunity for the user to interact with the module about how to utilize those notes. It also provides a data browsing and inspection interface where filtering, data options and ordering capabilities are some of the mechanisms that help a user format and view data in natural ways.

Bibliography

[1] Harp, B., & Neches, R. *Notecards: An Everyday Tool for Aiding in Complex Tasks.* In **Proceedings of the Workshop on Architectures for Intelligent Interfaces: Elements and Prototypes.** Monterey, CA. March, 1988. pp. 287-304.

[2] MacGregor, R., & Bates, R. *The LOOM Knowledge Representation Langauge.* In **Proceedings of Knowledge-Based Systems Workshop**, 1987.

[3] Neches, R. *Enabling Technology for Constructing Integrated User-Support Environments.* In **Proceedings of the AAAI Spring Symposium Workshop on Intelligent Interfaces**, Stanford University, March, 1990.

[4] Szekely, P. *Template-Based Mapping of Application Data to Interactive Displays.* In **Proceedings of the User Interface Software and Technology Conference**, Snowbird, Utah, October, 1990. pp. 1-9.